MQTT移植说明：

1. 附件中包括了需要的文件

|  |  |
| --- | --- |
|  |  |

包括：

1. cJSON.c, cJSON.h，这两个文件是JSON编解码包，最初期，你可以不管它们
2. 其它文件是MQTT库，其中，MQTTLwIP.c是新的，MQTTClient.c是修改过的，MQTTLwIP是MQTT的LwIP/FreeRToS适配。MQTTClient原实现处理得很不好，改进过了。
3. 将这个包解压，并放入您的工程。
4. 需要较大幅度适应性修改的内容——MQTTLwIP
   1. 毫秒精度的时钟：#4 - #29，使用了HAL\_GetTick()，请改为你自己的函数。
   2. LWIP\_read: #30 - #58：原库函数设计是：从Socket中读指定字节，如果指定时间数据不足则超时。但实际底层lwip似乎没有很好实现这个功能，造成死等。当前实际实现，setsockopt并不起作用，recv读指定数量字节。在修改实现中，请视情况修改。
   3. checkWaitingPacket：#59 - #70检查是否有数据包在等待。使用了select函数，请视情况修改。（特别注意，如果不用select，在没有数据等待的情况下，请把CPU时间释放给其它线程）
   4. LWIP\_write: #93 - #103，setsockopt实际没有起作用，使用write函数，请视情况修改
   5. LWIP\_disconnect: #106 - #111，关闭套接字
   6. NewNetwork：#112 - #118，相关函数入口定义
   7. 创建TCP连接：#121 - #153，接口是ConnectNetwork，请根据情况修改，现阶段直接连IP地址，没有DNS查找
5. mywork.c是MQTT库的一个实用示例
   1. TCPClt：#174 - #218是主循环，实际是一个线程的执行函数，NewNetwork完成适配，ConnectNetwork检查TCP/IP栈是否就需并实施TCPIP连接，MQTTClient为MQTT连接准备空间（此处使用静态变量），MQTTConnect完成MQTT服务器登录，MQTTSubscribe注册感兴趣的话题（下行命令接收）,PublishLED实际通过MQTTPublish发送数据包，MQTTYield需要经常性调用以完成MQTT事务操作（例如检查是否有数据包等待处理，数据包的回应，KeepAlive的处理等）。MQTTDisconnect释放MQTT连接
   2. PublishLED是一个使用JSON格式封装数据并发送的函数
   3. messageArrived供回调。当收到数据包以后，本函数被调用。这儿示范了一个对JSON解包的过程（并没有根据最新的格式调整），仅供参考。
   4. #10-#28 包括了常量定义。实际使用中，需要把相关ID和Token改为从网站上申请的数据。缓冲区也在此定义，请根据实际使用大小调整，以免溢出。
   5. JSON相关函数从堆中申请内存，请保证堆中有充足空间。
6. 备注：这次的小包里面，MQTTLwIP和MQTTClient有小改动，注释掉了一些调试用的printf等。